

УДК 681.5:004.8
DOI: 10.15827/0236-235X.133.106-113

Дата подачи статьи: 28.07.20
2021. Т. 34. № 1. С. 106–113

Интеллектуальный подход к автоматизации технологических и производственных процессов

С.Ю. Рябов¹, старший инженер, sergey.u.ryabov@gmail.com
Ю.В. Рябов², к.т.н., доцент, ryabov_yuri_atp@mail.ru

¹ Fannie Mae 11600 American Dream Way, Reston, VA 20190, USA

² Уфимский государственный авиационный технический университет,
г. Уфа, 450000, Россия

Рассматривается подход к автоматизации производства, в частности, к автоматизированному проектированию технологических процессов. Обработка данных в существующих системах сводится к набору правил, а исполняющая программа в своей реализации напоминает конечный автомат. Очевидно, что такой подход имеет свой потолок. Предлагается представлять производственный процесс как нечто целое, описываемое интеллектуальной моделью.

Принятая модель автоматизации технологических и производственных процессов базируется на теории графов и графовом представлении данных и знаний. Граф рассматривается как некоторая функция времени и вычислений.

Предлагается использование суперграфа как набора абстрактных и определенных заданных узлов и абстрактных и статических отношений. Таким образом, каждый сценарий реального мира, каждая производственная ситуация, рассматриваемая в каком-либо масштабе, будет смоделирована как некий подграф суперграфа. Интеллектуальной платформой для реализации вычислений может быть Акка, представляющая собой реализацию акторной вычислительной модели. Она позволяет осуществлять интеллектуальный подход к решению задачи автоматизации производственных и технологических процессов.

Рассмотрен пример построения части суперграфа для механообработки типовым переходом элемента детали с помощью соответствующего инструмента и выбора режима обработки. Итогом работы такой системы будет граф с вершинами и отношениями, описывающими знания технологических операций или состояние производственного процесса. Результат может быть передан иной системе на исполнение, сохранен в БД или использован для анализа ситуации.

Ключевые слова: производственный процесс, технологический процесс, граф, суперграф, интеллектуальная платформа, статический узел, узел данных, абстрактный узел, вычислитель.

В настоящей статье авторы рассматривают попытку использовать интеллектуальный подход к решению задачи автоматизации производственной сферы несколько иначе по сравнению с существующими системами. Большинство современных решений основано на общеизвестных и широко распространенных принципах. В качестве примера можно привести два класса систем:

– CAD/CAM-системы, применяемые для проектирования изделий и создания управляющих программ обработки деталей на станках с ЧПУ [1];

– PDM/PLM-системы, обеспечивающие управление всей информацией об изделии и жизненным циклом продукции.

Принимая во внимание ресурсы, затраченные на развитие процессов автоматизированного проектирования, разнообразие программных продуктов и степень охвата ими всего

производственного процесса, возникает вопрос об отсутствии до сих пор решения, которое охватывало бы весь производственный процесс в целом, было бы достаточно гибким, чтобы привносимые изменения не усложняли решение до такой степени, когда его поддержка экономически неэффективна, предоставляло бы возможность относительно простого интегрирования с существующими решениями.

PDM/PLM-системы являются своего рода расширенной БД для CAD/CAPP/CAM-систем. Иногда они пересекаются друг с другом, но в основном такие решения выглядят фрагментарно и изолированно. Если их все проанализировать, то видно, что они – продукт формализации части предметной области.

Формализованная схема, описывающая некоторую модель действительного мира, выражена в большинстве случаев некоторой декларативной моделью, точнее – модель хранения

данных представлена в виде реляционной модели или объектной модели.

PDM/PLM-системы представляют собой набор бизнес-правил, например, правил определения данных об изделиях, правил сохранения и поиска технологической документации, учета инструмента на складе и т.д. Это стандартный подход, в большинстве случаев работающий там, где возможна четкая формализация процессов и данных. До какой-то степени это напоминает экспертную систему, основанную на правилах, с одной стороны, и конечный автомат, с другой. Иными словами, обработка данных правил сводится к набору правил, а сама исполняющая программа в своей реализации напоминает конечный автомат.

Очевидно, что такой подход имеет свой недостаток. С увеличением количества правил система становится все более сложной или даже громоздкой, менее надежной и в результате неэффективной. Также CAPP-системы в настоящее время не позволяют осуществлять принятие решений по изменению технологических параметров процесса с учетом информации о ходе процесса и о его результатах (параметры изделия) [2].

Следует отметить, что данная проблема в значительной степени относится к PDM/PLM- и CAPP-системам, так как CAD/CAM-системы представляют собой удобный графический редактор, вычислительные модули и библиотеки различного назначения, то есть являются в основном системами моделирования, а не анализа и применения правил.

Постановка задачи

Авторы статьи предлагают представлять производственный процесс как нечто целое, описываемое интеллектуальной моделью. Подход к моделированию оценивается с точки зрения способности предлагаемой теории или подхода представить процессы, явления действительности в виде наиболее полной и непротиворечивой модели.

Модель можно считать полной, если она способна воспринять и обработать все возможные данные из моделируемой предметной области, воспроизводя при этом результат, соответствующий представлению о действительности. Если существуют такие данные, при которых модель не возвращает результат или результат не соответствует действительности, модель нельзя считать полной.

Модель можно считать непротиворечивой, если она не воспроизводит противоречащие действительности результаты и два противоречивых результата на основании одних и тех же данных.

Данные свойства являются лишь аналогами таких же свойств формальных систем. Доказательство непротиворечивости – весьма сложная задача даже для относительно простых формальных систем. Очевидно, для модели подобное доказательство займет намного больше ресурсов и времени. Данные свойства можно сформулировать следующим образом.

С формальной точки зрения существуют два типа полноты модели – семантическая и синтаксическая. Семантическая полнота модели означает, что все истинные утверждения, сформулированные на языке теории, могут быть доказаны, а синтаксическая полнота – что добавление еще одной аксиомы к системе ведет к ее противоречивости [3]. Применительно к системе данные свойства можно объяснить следующим образом: система, реализованная в виде программного кода, может найти решение и воплотить его в жизнь с использованием ресурсов, заданных при постановке задачи, что отражает семантическую полноту. Для процесса производства это сводится к тому, что технологический процесс будет включать в себя только уже имеющиеся на заводе операции, инструмент, оснастку, оборудование и заготовки.

Синтаксическая полнота состоит в том, что набор функций, созданный для принятия решения, не ведет к неоднозначным и, возможно, противоречивым результатам. Непротиворечивость системы сводится к утверждению, что в системе нельзя вывести A и $\sim A$ [3]. Применительно к программной системе, используя тот же пример технологического процесса, можно сказать, что система не должна давать два различных результата на основании одних и тех же данных. Этот пример довольно абстрактный, и свойство непротиворечивости в данном случае не стоит толковать на таком простом уровне. Чтобы сделать его более понятным, следует сказать, что на основании классов данных, где данные могут варьировать до определенной степени, система должна генерировать класс решений, которые будут ковариантны начальным данным, если же свойство ковариантности нарушится, то такая система моделирования (принятия решения) будет противоречивой. Конечно же, здесь важно определить степень варьирования данных и задать критерии ковариантности.

Стоит отметить, что модель – это всегда какое-то приближение к реальному миру, но никогда не его точное представление. Строго говоря, не существует полных и непротиворечивых систем.

С учетом сформулированных правил определим принципы системы:

- представление предметной области должно быть основано на атомарных элементах;
- отношения между этими элементами образуют более сложную конструкцию;
- хранение, поиск данных, изменение модели должны быть унифицированы и поддаваться формальному описанию;
- модель должна быть открытой для добавления новых элементов, замены, но не изменения существующих элементов.

Авторы статьи взяли за основу такой модели теорию графов и графовое представление данных и знаний.

Решение

Выбор графового представления обусловлен тем, что графы состоят из узлов и отношений между ними, узел графа может быть либо

неким абстрактным, либо вполне определенным объектом. Отношения между узлами могут быть статическими, возникшими в момент описания объекта моделируемой области либо результатом развития модели во времени. В общем, у графа нет схемы, которая в данном случае была бы, скорее, недостатком модели. Существуют теории поиска данных на графе и их реализации [4, 5]. Сформулируем модель глубже, а граф рассмотрим как некоторую функцию времени и вычислений (см. таблицу).

Суперграф

Вычисление можно назвать процедурой или функцией. Каждое вычисление должно иметь фиксированное описание – сигнатуру функции, то есть функция может принимать определенное число аргументов и возвращать результат или несколько результатов.

В рассматриваемой модели функция может принимать от одного до нескольких аргументов и каждый аргумент – это узел графа. Соответственно, функции могут произвести новый узел графа, что означает успешное выполнение, а могут и не произвести новый узел – неуспешное выполнение функции. Также оче-

Определения графа с позиции функции времени и вычислений

Graph definitions from the position of the time function and calculations

Определение	Свойства	Ось вычислений	Ось времени
<i>N</i> – узел (node)	Узлы графа могут быть абстрактными и определенными. <i>Абстрактные узлы (АУ)</i> существуют исключительно в суперграфе. Определенные узлы создаются в момент развития графа	Элементы оси – последовательные вычисления, применяемые к узлам. В результате вычисления в узлах создается или не создается новый узел	Ось времени – последовательность событий, приходящих в систему. За момент времени в систему могут прийти несколько событий. Каждое событие представляется в виде узла
<i>E</i> – отношение (edge)	Статические отношения хранятся в БД как описание некоторой статической части модели. Динамические отношения создаются в момент обработки входящих данных и построения графа	Отношения выстраиваются в результате применения вычисления	Ось времени в отношениях между узлами: узел – событие, применение вычисления к узлу-событию создает или не создает отношения
Граф	Граф представляет модель реального мира. Он имеет статическую часть, описывающую статическую часть модели, и динамическую, появляющуюся в момент применения вычислений	Вычисления, применяемые к узлам графа, могут создать, а могут и не создать новый узел графа	Граф изменяется во времени, в каждый момент времени события реального мира появляются новые динамические узлы графа, а применение вычислений к событиям изменяет граф

видно, что данный подход детерминирует структуру графа. Иными словами, динамические узлы, произведенные применением функций, могут иметь различные атрибуты, но их типы и комбинации конечны, также конечно множество отношений между этими узлами. Какими бы большими эти комбинации и множества отношений ни были, их число конечно и определено применяемыми функциями. Таким образом, этот факт делает модель преддетерминированной и все вычисления всего лишь детерминантами данного графа или, скорее, уже суперграфа.

Суперграф – это граф абстрактных и определенных заданных узлов и абстрактных и статических отношений. Каждое новое событие или группа событий, приходящих в один и тот же квант времени, становится частью графа, и к нему применяются вычисления, которые, в свою очередь, могут породить узлы, изменить состояние АУ и сделать данный узел определенным. Таким образом, каждый сценарий реального мира, каждая производственная ситуация, рассматриваемая в каком-либо масштабе, будут смоделированы как некий подграф суперграфа. Развитие модели может быть расширено в представление узла как некоего подграфа, видимого из основного графа как нечто целое и неделимое. Очевидно, для этих целей потребуется определенного рода функтор, который должен сворачивать граф в один узел. Как это сделать и как описать подобный функтор – задача дальнейшего исследования.

Каждая модель должна быть каким-либо образом описана. В данном случае описание модели есть описание суперграфа. Как было сказано выше, суперграф может состоять из очень большого числа узлов и отношений между ними. Иными словами, это будет множество всех возможных успешных вычислений на всех возможных комбинациях данных. Получить описание такого графа трудно. Однако в этом нет необходимости – достаточно задать точное описание функций, тогда свод правил и описание функций в виде формул и будет описанием модели.

Описание графа вполне может быть выражено математическими определениями. Обе структуры можно рассматривать как разновидность семантических сетей [6]. И, хотя модель не повторяет один в один структуру сети, математические формулировки модели будут подобны тем, которые используются для семантических сетей. Семантическая сеть определена как графовая структура для представле-

ния знаний через шаблоны взаимосвязанных узлов [7].

Из всех категорий семантических сетей графовая модель типа «исполняемые сети» наиболее соответствует описанию суперграфа, содержащего в себе процедуры трансформации, способные изменять саму сеть. В таких сетях имеются маркеры и сообщения, которые как данные передаются от узла к узлу, а процедуры производят вычисления внутри узла. В таких сетях применяются также преобразования графов, объединяющие графы, модифицирующие и разбивающие их на более мелкие графы [8]. Преобразования осуществляются внешними по отношению к графам программами. Программы запускаются самими графами и ведут себя как химические реакции, объединяющие молекулы или разбивающие их на части. К исполняемым сетям относятся также сети Петри, которые имеют пассивные и активные узлы, называемые переходами [9]. Кроме того, они имеют набор правил для маркировки мест выполнения или запуска переходов.

Семантические сети уже использовались в некоторых областях человеческой деятельности, например, Пирс увидел подобие между понятием валентности и графическим представлением молекулы вещества. Графовые представления знаний зачастую сводятся к графическому представлению логики первого порядка.

Реализация модели

В настоящее время существует довольно много различных БД, которые хранят данные в графовом представлении, например, Neo4j, Gremlin, ArangoDB. Любая из этих платформ может быть использована для хранения графов и поиска данных на графах. Известным и принятым де-факто стандартным языком поиска данных на графах является Gremlin – доменный язык для обхода графов и формирования запросов в графовой БД, проведения анализа и манипуляций с данными [10]. Остается вопрос относительно интеллектуальной платформы для реализации вычислений. Как уже было сказано, графовая модель по сути своей является одной из реализаций семантической сети, поэтому любая платформа, подходящая для реализации семантической сети, подойдет и для реализации интеллектуальной модели. Данная платформа должна поддерживать

– создание узлов графа как изолированную и неизменяемую структуру данных;

- описание отношений между *статическими узлами (СУ)*;
- создание новых узлов посредством применения вычислений.

Узлы и вычисления по своей природе должны быть распределяемыми по вычислительным ресурсам (JVM, VM и т.д.). Узлы графа должны быть неизменяемыми (immutable), но заменяемыми.

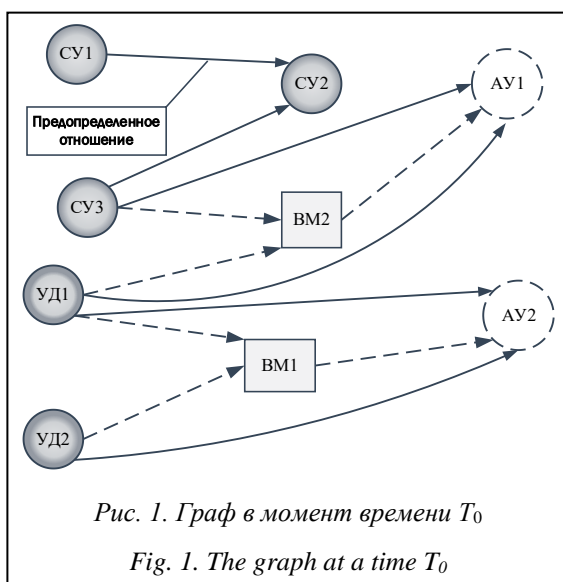
С учетом всех параметров такой платформой может служить Akka [11], которая представляет собой реализацию акторной вычислительной модели и позволяет реализовать интеллектуальный подход к решению задачи автоматизации производственных и технологических процессов.

Реализованная система будет работать при следующих условиях.

1. Имеется система акторов, которая может принимать определенные *узлы данных (УД)* и строить из них новые.
2. Приложение обладает функцией загрузки графов или частей большого графа в акторы-данные (aka Persistence-Actors) из БД.
3. Один или несколько акторов-диспетчеров будут управлять вычислительным процессом.

Тогда весь процесс построения графов состояний производственного процесса или производственных технологических решений будет сводиться к схеме, изображенной на рисунке 1.

Существуют СУ-х, определяющие статические данные и начальные условия. Применительно к производственной области к СУ можно отнести оборудование, инструмент,



условия обработки. Входная информация определяется УД-х. Это информация о заготовках, о новых деталях, обновлениях технологических процессов и оборудования.

СУ могут иметь предопределенные отношения. Входные данные (УД-х) вместе с СУ-х служат входными данными для *вычислительных модулей (ВМ-х)*. На уровне реализации посредством Акка графовая модель выглядит следующим образом: специальные агенты-акторы рассылают данные СУ и УД как сообщения ВМ-акторам, которые собирают необходимые узлы и на основании данных могут создавать или не создавать новый УД.

Следует отметить, что ВМ-акторы могут создать только предопределенные УД. Такая детерминированность ведет к тому, что ВМ-акторы могут создать новые УД и отношения между существующими и новыми УД. Так как до момента поступления данных неизвестно, будет порожден новый узел или нет, все потенциальные новые УД суперграфа помечаются как АУ-х.

Таким образом, суперграф содержит три типа узлов и два типа отношений: СУ (существующие с момента появления системы), узлы входных данных и АУ данных – потенциальные УД, которые могут быть созданы посредством ВМ-акторов. Отношения между узлами могут быть статическими (заданными в момент создания системы) и созданными ВМ-акторами: между СУ, УД и вновь созданными УД (СУД), которые перешли из категории АУ в категорию реальных.

Примером могут служить СУ3 и УД1, отправленные в ВМ2-актор. ВМ2 создает новый УД из АУ1 и связывает отношением СУ3 и АУ1 (рис. 1), после чего узел АУ1 превращается в сгенерированный УД – ГУД1 (рис. 2). Этот процесс продолжается отправлением новых узлов (ГУД1) ВМ3-актору и т.д. (рис. 2).

Суперграф, таким образом, получил свое название именно потому, что он может описывать все возможные графы информационных процессов производства различных изделий. Каждый раз при завершении обработки входных данных реализуется только часть графа. Остальные узлы так и остаются абстрактными.

Рассмотрим пример построения части суперграфа для механообработки типовым переходом элемента детали с помощью соответствующего инструмента и выбора режима обработки (рис. 3).

Пусть СУ1 – инструмент обработки внешней поверхности детали, СУ2 – мерительный

инструмент для элемента детали типового перехода, УД2 – данные элемента детали, УД1 – данные заготовки. ВМ1-актор определяет режимы обработки исходя из типа инструмента и типа заготовки. При этом генерируется УД по режимам обработки (ГУД1). ВМ2-актор формирует выполняемый переход для выбранного элемента детали, и ГУД2 определяет тип перехода и получаемые размеры на основе данных заготовки (УД1), данных инструмента (СУ1), данных элемента детали (УД2), данных мерительного инструмента (СУ2) и сгенерированных режимов обработки (ГУД1).

На основе данных элемента детали (УД2) и сгенерированного перехода (ГУД2) ВМ3-актор формирует программу обработки в виде АУ1. После параметризации АУ-х по всем переходам выполняемой операции узлы переходят в категорию реальных.

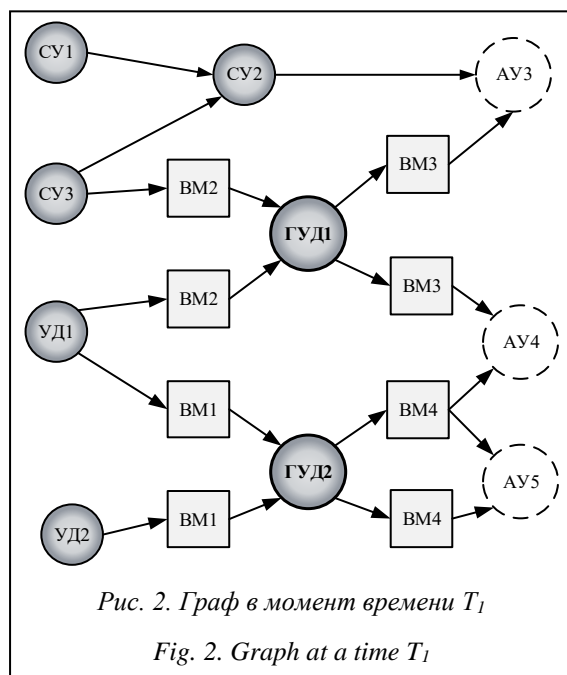


Рис. 2. Граф в момент времени T_1

Fig. 2. Graph at a time T_1

Система загружает статические данные из БД, в том числе СУ и предопределенные отношения. Запускает акторы данных, которые загружают из БД свою часть графа. Принимает данные извне: например, задание разработать технологическую операцию.

Начинается исполнение задания: акторы-вычислители получают сообщения, в которых указаны часть графа, ссылка на узлы графов, к которым нужно применить вычисление.

Если вычисление было успешным, создается новый УД, который сохраняется в графе

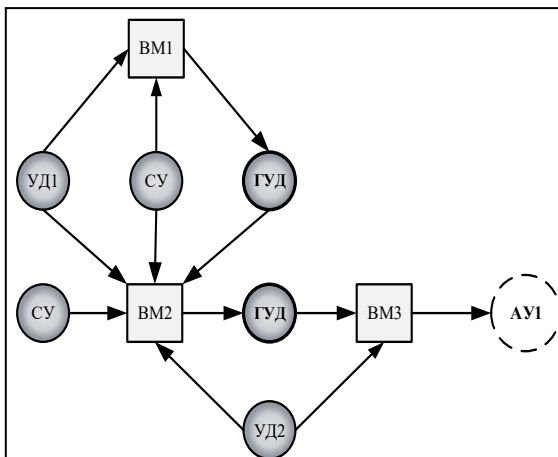


Рис. 3. Граф механообработки элемента детали

Fig. 3. The graph of machining of a part element

через актор данных; если вычисление не принесло результатов (ничего не происходит), граф не меняется.

Управляет всеми вычислительными процессами диспетчер – это актор, позволяющий проверять готовность задания для выяснения, нужен ли следующий вычислительный шаг; выявлять акторы-вычислители, которые можно применить; выделять для них узлы или часть графа и посылать сообщение с этими данными.

Процесс завершается, когда либо задание выполнено (нормальное завершение), либо все возможные вычисления завершены, а решения нет (ситуация нехватки знаний).

Заключение

Результатом работы системы будет граф с вершинами и отношениями, описывающими знания технологических операций или состояние производственного процесса. Такой граф-результат может быть передан иной системе на исполнение, сохранен в БД, использован для анализа ситуации. Этот же граф объединит ресурсы, описывающие производственный процесс и поведение всех его частей, соединяя их отношениями. Таким образом, можно реализовать новый интеллектуальный подход к автоматизации технологических и производственных процессов на основе обучения акторной вычислительной модели и применения технологических знаний, связанных с управлением производственными процессами.

Литература

1. Ловыгин А. Искусственный интеллект или хайп для станков // ПланетаСАМ. 2019. № 15. С. 54–55. URL: http://planetacam.ru/articles/high-tech/AI_CNC_machines/ (дата обращения: 28.05.2020).
2. Лютов А.Г., Рябов Ю.В., Шайдуллин Р.И., Шамбазов И.И. Интеллектуальное управление процессами технологической подготовки машиностроительного производства // Вестн. ЮУрГУ. Сер.: Компьютерные технологии, управление, радиоэлектроника. 2017. Т. 17. № 3. С. 117–124. DOI: 10.14529/СТСР170312.
3. Ильин В.В. История и философия науки. М.: Проспект, 2019. 335 с. DOI: 10.31085/9785392288359-2019-336.
4. Робинсон Я., Вебер Д., Эфрем Э. Графовые базы данных: новые возможности для работы со связанными данными; [пер. с англ.]. М.: ДМК Пресс, 2016. 256 с.
5. The Neo4j Operations Manual v3.5: URL: <https://neo4j.com/docs/operations-manual/3.5/> (дата обращения: 12.04.2020).
6. Андриченко А. Принципы построения семантических MDM-систем // САПР и графика. 2011. № 5. С. 69–73.
7. Chein M., Mugnier M.-L. Graph-based Knowledge Representation. Computational Foundations of Conceptual Graphs. Springer London, 2009, 425 p. DOI: 10.1007/978-1-84800-286-9.
8. Martin F.R. How Important Are Semantic Networks In Artificial Intelligence. URL: <https://analyt-icsindiamag.com/semantic-networks-ai/> (дата обращения: 26.01.2020).
9. Мальков М.В., Малыгина С.Н. Сети Петри и моделирование // Тр. КНЦ РАН: Информационные технологии. 2010. № 3. С. 35–40.
10. Apache TinkerPop. The Gremlin Graph Traversal Machine and Language. URL: <https://tinkerpop.apache.org/gremlin.html> (дата обращения: 07.02.2020).
11. Andr n J., Knight R. Reactive Programming with Akka. 2017. URL: <https://dzone.com/refcardz/reactive-programming-akka?chapter=1> (дата обращения: 12.10.2020).

Software & Systems
DOI: 10.15827/0236-235X.133.106-113

Received 28.07.20
2021, vol. 34, no. 1, pp. 106–113

The intelligent approach to automation of technological and production processes

*S.Yu. Ryabov*¹, Senior Software Engineer, sergey.u.ryabov@gmail.com

*Yu.V. Ryabov*², Ph.D. (Engineering), Associate Professor, ryabov_yuri_atp@mail.ru

¹ Fannie Mae 11600 American Dream Way, Reston, VA 20190, USA

² Ufa State Aviation Technical University, Ufa, 450000, Russian Federation

Abstract. The paper considers the approach to production automation, in particular, to the automated design of technological processes. Data processing in existing systems is reduced to a set of rules, and the executing program in its implementation is like a state machine. Obviously, this approach has its own ceiling. It is proposed to represent the production process as something whole, described by an intellectual model. The adopted model of automation of technological and production processes is based on graph theory and graph representation of data and knowledge. The graph is considered as some function of time and computation. It is proposed to use a supergraph as a set of abstract and defined given nodes and abstract and static relations. Thus, every script of the physical reality, every manufacturing situation, considered at any scale, will be modeled as a subgraph of a supergraph. Akka, which is an implementation of an actor computational model, can be an intelligent platform for the implementation of computations. It allows for an intelligent approach to solving the problem of automating production and technological processes.

An example of constructing a part of a supergraph for machining a part element is considered by a typical transition-side, including the corresponding tool, processing modes, and a measuring tool. The result of such a system will be a graph with vertices and relations describing the knowledge of technological operations or the state of the production process. The result can be transferred to another system for execution, saved in the database, or used to analyze the situation.

Keywords: production process, technological process, graph, supergraph, intelligent platform, static node, data node, abstract node, calculator.

References

1. Lovygin A. Artificial intelligence or hype for machine tools. *PlanetaCAM*, 2019, no. 15, pp. 54–55 (in Russ.). Available at: http://planetacam.ru/articles/high-tech/AI_CNC_machines/ (accessed May 28, 2020).
2. Lyutov A.G., Ryabov Yu.V., Shaydullin R.I., Shambasov I.I. Intellectual control of processes of technological preparation of machine-building production. *Bull. of SUSU. Computer Technology, Control, Radio Electronics Series*, 2017, vol. 17, no. 3, pp. 117–124 (in Russ.). DOI: 10.14529/CTCR170312.
3. Ilyin V.V. *History and Philosophy of Science*. Moscow, 2019, 335 p. DOI: 10.31085/9785392288359-2019-336.
4. Robinson I., Webber J., Eifrem E. *Graph Databases: New Opportunities for Connected Data*. O'Reilly Media, 2015, 238 p. (Russ. ed.: Moscow, 2016, 256 p.).
5. *The Neo4j Operations Manual v3.5*. Available at: <https://neo4j.com/docs/operations-manual/3.5/> (accessed April 12, 2020).
6. Andrichenko A. Principles for building semantic MDM systems. *CAD and Graphics*, 2011, no. 5, pp. 69–73.
7. Chein M., Mugnier M.-L. Graph-based Knowledge Representation. *Computational Foundations of Conceptual Graphs*. Springer, London, 2009, 425 p. DOI: 10.1007/978-1-84800-286-9.
8. Martin F.R. *How Important Are Semantic Networks In Artificial Intelligence*. Available at: <https://analyticsindiamag.com/semantic-networks-ai/> (accessed January 26, 2020).
9. Malkov M.V., Malygina S.N. Petri nets and modelling. *Transactions of Kola Science Centre of the RAS: Information Technologies*, 2010, no. 3, pp. 35–40.
10. *Apache TinkerPop. The Gremlin Graph Traversal Machine and Language*. Available at: <https://tinkerpop.apache.org/gremlin.html> (accessed February 07, 2020).
11. Andr n J., Knight R. *Reactive Programming with Akka*. 2017. Available at: <https://dzone.com/ref-cardz/reactive-programming-akka?chapter=1> (accessed October 12, 2020).

Для цитирования

Рябов С.Ю., Рябов Ю.В. Интеллектуальный подход к автоматизации технологических и производственных процессов // Программные продукты и системы. 2021. Т. 34. № 1. С. 106–113. DOI: 10.15827/0236-235X.133.106-113.

For citation

Ryabov S.Yu., Ryabov Yu.V. The intelligent approach to automation of technological and production processes. *Software & Systems*, 2021, vol. 34, no. 1, pp. 106–113 (in Russ.). DOI: 10.15827/0236-235X.133.106-113.